BSc Computer Science
Design project group 22

Smart Library: Monitoring the Monitors

Trivedi. A, Abdul Wahab. A, Thakurani. K, Botez. R, Trifonova. R

Supervisor: Faiza Bukhsh

April, 2024

Department of Computer Science
Faculty of Electrical Engineering,
Mathematics and Computer Science,
University of Twente

UNIVERSITY OF TWENTE.

Contents

Abstract

This report details the development of a library management system at the University of Twente, aimed at enhancing efficiency and sustainability by managing monitor area overcrowding. Utilizing recently installed sensors, the project integrates sensor data into two dashboards: a Management Dashboard for staff with power consumption analytics and a User Dashboard showing real-time monitor availability for students. The system reduces unnecessary library visits and improves student productivity by optimizing space utilization and providing energy usage insights to support sustainability initiatives. Future enhancements could broaden the system's integration across campus and expand its scope to include additional study areas.

Chapter 1

Introduction

The main objective of this project is to enhance the student experience at the University of Twente by offering mitigation to the overcrowding problem at the university library through smart technologies and data analytics.

Libraries are vital centers for learning and research, adapting to the evolving needs of students in the digital era. Yet, they face new challenges, including efficiently managing space and resources. One such challenge is the overcrowding of monitor areas, where students often seek additional screens for their studies.

The University of Twente library encounters this issue, with the monitor area often being full during peak hours. This leads to frustration and inconvenience for students, who may make unnecessary trips to the library only to find out that all spots are occupied, disrupting their plans and productivity.

To tackle this problem and improve the student experience, the proposed solution uses smart technologies and data analytics. This solution comprises two main parts: a Management Dashboard and a User Dashboard. Their main purpose is to inform the users about the availability of the monitor area inside the library of the University of Twente.

The Management Dashboard provides the library staff with insights into the monitor area usage. By integrating data from motion sensors attached under the desks, and smart plugs connected to the monitors, it offers real-time monitoring and trend analysis. This aids the staff in decision-making for resource allocation, space management, and maintenance.

The User Dashboard assists students by providing real-time information on monitor availability. Accessible through an online platform such as MazeMap, it enables students to plan their study sessions more effectively by providing an understanding of the monitor area occupancy.

By implementing these solutions, the aim is to alleviate monitor area overcrowding, enhance efficiency, and improve the overall student experience. Through smart technology and data-driven insights, the goal is to create a more responsive and student-focused library environment. Furthermore, along the journey of completing the project, we realized that our goals had shifted. We were no longer solely focused on planning, but we helped the library achieve its sustainability goals as well.

The report is divided into 8 chapters the first of which, Chapter 2, provides an analysis of existing solutions and their effectiveness. Chapter 3 gives a detailed overview of the proposed system. Following, Chapter 4 discusses the functional and non-functional requirements formed and prioritizes them. Chapter 5 discusses the proposed architecture and justifies the design decisions. Chapter 6 outlines the testing strategies incorporated in the development phase. Chapter 7 provides the result of the project, a reflection, and future improvements. The final chapter,Chapter 8, provides the actual planning that took phase throughout each week, team responsibilities, and stakeholders' satisfaction regarding the final result. A brief summary of the system is shown below:

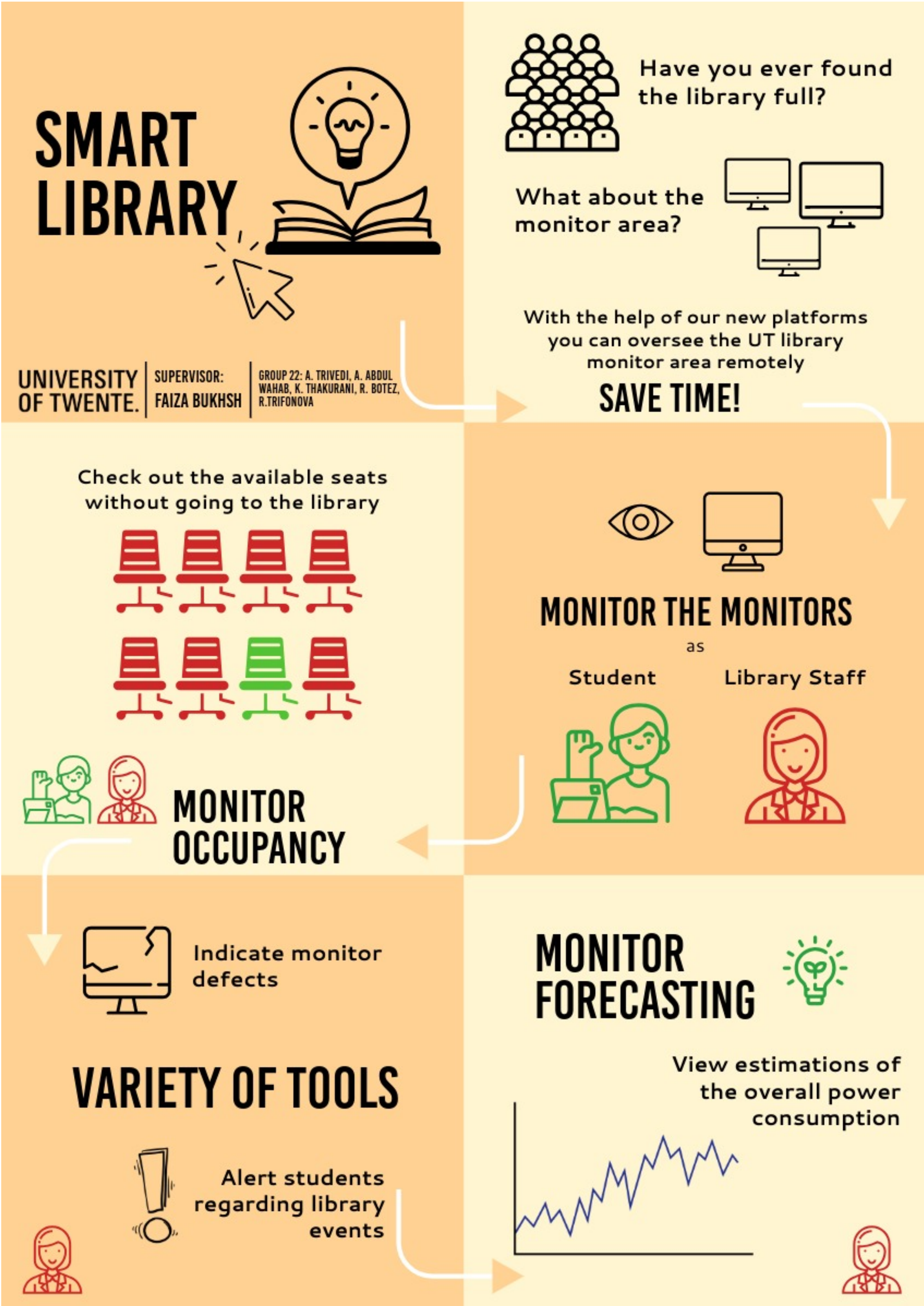

FIGURE 1.1: System overview

Chapter 2

Domain Analysis

In this chapter, the process of identifying the domain of the system is discussed. The view of the client and users, the software environment used, and the application of other UT software are mentioned in this section. Domain analysis is necessary to produce an unambiguous definition in the form of specific techniques [2]. It is a well-defined set of guidelines and metrics which enables the development, reuse, and testing of software. Domain analysis aids the planning for future development.

2.1 Introduction to Domain

The domain in which the system concerns the overcrowding of the monitor area in the library. The overcrowding of monitor areas in the library poses several challenges for both students and librarians. Students struggle to find available monitor spaces, leading to frustration and inefficiency in their study routines. Additionally, librarians face difficulties while managing the allocation of monitor spaces and planning for future infrastructure enhancements. The system should address these challenges in a comprehensive solution that combines real-time monitoring, data analysis, and user-friendly interfaces to facilitate efficient space utilization. The system must empower students and librarians to make the most of available resources and conduct a learning environment for all.

2.2 General Knowledge of the Domain

Within the University of Twente library, a diverse array of sensors has been implemented in the monitor area, comprising smart plugs and motion sensors. These sensors operate distinctively, necessitating the system to accommodate both data types seamlessly. The goal of digitalization is not only for the students and librarians to know the status of each monitor at any given time, but can branch out to different topics of management. Such topics are the necessity of supplementing the number of seats with monitor access in the future, increases in the frequency with which the cleaning service providers are cleaning the space, and increases in investments such as better chairs due to patterns that indicate prolonged continuous study sessions in the monitor area. Understanding and visualization of these sensors can determine whether the system can successfully be used.

2.3 Client, Users and Interested Parties

Two categories of clients and users would benefit from this project: students and librarians.

Students of the University of Twente often use the library for the silent study spaces the library can provide. The library's schedule is an extended one, stretching outside working hours, throughout the weekend, and sometimes throughout holidays. This attracts students to spend long parts of their day in the library, especially during exam weeks, which can cause the library to become very crowded. Along with other facilities, such as individual or project study rooms, the monitors' area is one of the most sought-after study places in the entire library. This study space consists of several monitors that students can connect their laptops to. There is no time limit, nor a booking system for using these monitors, making it a first come, first served system. Often, students come to the library solely for these monitors, just to find out all of them are taken when they arrive at the library.

Librarians are employees of the University of Twente who perform various duties around the library. One of them is the service desk, where students can also come to ask questions. This is also where the librarian can supervise the activity and the foot traffic inside the library, and take appropriate measures if needed. Besides the visual observation of the monitor area and its occupancy level, there is not much else the librarian can do about how the monitors are being used. One of the issues they encounter is seat abandonment. This happens when students abandon their personal belongings for extended periods, in rare instances up to a few hours, to reserve in an unofficial way that respective spot. This prevents other students from using that spot for the duration, which creates inefficiency and overcrowding. Librarians have no way of knowing exactly which spot has been abandoned for how long, nor do they know what the monitor area occupancy level would look like for the rest of the day. Having such software will help librarians identify such occurrences and take appropriate measures to prevent them from happening again in the future.

2.4 Software Environment

The client did not have any requirements concerning the technology stack. Therefore, the choice was made to utilize technologies that are common to use in this kind of project. One of them is Flask, often used as a web framework. The later sections will discuss in detail the technologies and the choice of using them.

The smart plugs serve as sensors installed between the monitors and their power sources, tasked with capturing data, such as the power consumption of the monitors. During standby mode, the monitors draw a minimal amount of energy, sometimes equivalent to zero, which is highly dependent on the monitor's model and age. However, significantly higher energy consumption can be anticipated when the monitors are actively in use performing demanding tasks, operating at maximum brightness, and simultaneously charging connected laptops via ports such as USB-C, or other peripherals.

Ultrasonic motion sensors are placed under the desk of each monitor and have the role of detecting movements from the user of the monitor. This sensor helps with understanding if a student is currently using the seat, or if only the laptop is connected to the monitor, and the seat has been abandoned. The sensor returns a variable of type boolean equal to "true" if the seat is occupied, or "false" if the seat is free.

MazeMap is an interactive map that students can use to locate rooms across the campus. One of the areas available for viewing is the library, where the monitor area is among the regions providing additional information about the placement of each desk along with a unique identification code. Presently, the system does not provide information regarding the occupancy of the monitors; it only indicates the location of this area inside the library and the placement of each desk in its designated area.

2.5 Procedures of the current situation

In the existing scenario at the University of Twente, there is a lack of dedicated systems for visualizing monitor occupancy within the library premises. With the recent deployment of sensors, after recognizing their technological potential, the library has taken proactive steps by launching this project. Its initiative aims to explore the capabilities of such sensors and identify valuable information that can benefit both students and library administrators for taking steps in the right direction; with values such as technological advancement and sustainability in mind.

The project's objectives extend beyond mere sensor deployment; it seeks further into the data collected by these sensors to find meaningful patterns and insightful trends regarding monitor usage within the library. By leveraging the data captured by the sensors, the project aims to address real-time visualization of monitor occupancy. Additionally, it helps to understand how this information can be effectively utilized to enhance the overall library experience for students and optimize resource allocation from the library's perspective.

Through this project, the library is developing a culture of innovation and technological integration within its operations. By embracing data-driven decision-making, the library administration aspires to create a more dynamic and responsive environment. Furthermore, the project serves as a testament to the university's commitment to leveraging cutting-edge technology to improve the academic experience and a conducive learning environment for its students.

2.6 Integration to other UT Software

The library of the University of Twente utilizes an internal project called "The People Counter". Consisting of a set of motion sensors placed inside the door frames at the library's entrance, and a dashboard visible at the service desk. The goal of the system is to provide the librarian with a clear idea of the space availability of the entire library area inside the building. One of the main information points that this dashboard provides is the number of people currently in the library; this number can be used to predict the occupancy at the monitor area. In principle, the ratio of people entering the library and sitting at the monitor area can be calculated with the aid of the sensors inside the door and the two sensors in the desk area. This ratio can be used to fit a model to predict occupancy for n-time intervals.

The library has numerous endpoints to send and collect data collected by the sensors placed around campus, not only the library. As a result of this, access to the smart plugs and motion sensors was required for a clear understanding of the monitor area. Sensors like the two mentioned before (smart plugs and motion sensors) are also in other campus buildings such as Ravelijn, therefore, data cleaning is essential since all these sensors have unique IDs and all of them go in the same database, where they get collectively stored.

The database that the library uses contains the readings coming from all these sensors. A table containing the list of IDs for the smart plugs, and one for the IDs of the motion sensors was provided by the library's staff to aid with the identification of each sensor and its data according to the real-life placement. The monitor area is a fixed space, where the monitors cannot be moved around. Therefore, to connect the sensors to real-life placement, a sketch visualization of the monitors and their placement, together with the IDs of their respective sensors, was also provided by the library's staff.

2.7 Conclusion

In conclusion, the implementation of a system to address the overcrowding of monitor areas within the University of Twente's library presents an opportunity to enhance the learning environment for students and improve operational efficiency for librarians. By leveraging sensor technology and data-driven insights, the proposed system aims to provide real-time visibility into monitor occupancy, empowering users to make informed decisions regarding space utilization. Furthermore, integration with existing university software, such as "The People Counter," demonstrates a commitment to leveraging available resources and fostering collaboration across departments. Moving forward, the successful deployment of this system has the potential to significantly impact library management practices and contribute to a more seamless and productive academic experience for all stakeholders involved.

Chapter 3

System proposal

In this chapter, the system proposal to the client is explained by discussing different proposal phases during the project. In chronological order, these proposals are a requirement proposal, a mock-up proposal, a system introduction, a usability testing phase, and a final presentation of the system to the client.

3.1 Requirements proposal

At the beginning of the design project, one of the most important requirements was to plan meetings with the stakeholders. Weekly meetings were held to update the library's team about the progress. The meetings were attended by the product owner, the library's management staff, and a technical person from LISA. These meetings helped the team in defining the scope of the project and the requirements the client had for the system. After a few sessions, once most requirements were laid out and verified by the stakeholders, a new phase of the project started. New requirements were proposed by the client during the agile development process.

3.2 Mock-ups proposal

In response to the challenge of overcrowding the monitor area during peak hours inside the University of Twente's library, the team proposed a comprehensive two-stage solution. This solution addresses the needs of both library management and students, enhancing the overall experience and efficiency of monitoring usage within the library.

3.2.1 Management dashboard

The first stage of the solution involves the implementation of a "Management Dashboard", customized to the needs of the library's administrators and staff. Figure 5.7 can be referred to see the initial design for the management dashboard. This dashboard serves as a centralized platform for accessing insights and analytics related to the monitor area. Key components of the Management Dashboard include:

- Integration of sensor data
  Data from sensors installed on displays and smart plugs within the library's area will be integrated into the existing library management software. This integration ensures that all relevant data points are captured and accessible through a unified interface.

- Trend analysis
  By analyzing historical data collected from sensors, the Management Dashboard will identify trends and patterns in monitor usage. This deeper understanding of usage dynamics enables librarians to make informed decisions regarding resource allocation and facility management.

- User-friendly presentation
  Findings from data analysis will be presented in a user-friendly format within the Management Dashboard. Visualizations and reports will be designed to facilitate easy interpretation and decision-making for library management personnel.

- Maintenance and monitoring
  Regular checks will be conducted to ensure the proper functioning of sensors. Any issues identified will be addressed promptly to maintain data accuracy and reliability. Additionally, a system for continuous monitoring of sensor health will be implemented to detect and resolve any potential issues proactively.

3.2.2 User dashboard

The second stage of the final solution focuses on providing a "User Dashboard" designed specifically for students. This dashboard offers real-time information on monitor availability within the library, allowing students to make informed decisions about when to visit. Key features of the User Dashboard include:

- Real-time availability
  Students can access real-time information on monitor availability, allowing them to plan their library visits more effectively and avoid unnecessary trips during peak hours.

- User-friendly interface
  The User Dashboard will feature an intuitive and easy-to-navigate interface, ensuring accessibility for all students regardless of technical proficiency.

- Integration with sensor technology
  Motion sensors installed under desks and smart plugs capable of detecting monitor usage will feed data into the User Dashboard, providing accurate and up-to-date information on monitor availability.

To summarise, the two-stage solution combines data analytics and user-friendly interfaces to address the challenge of monitoring area overcrowding in the University of Twente's library. By providing actionable insights for both library management and students, the aim is to improve the overall efficiency and experience of monitoring usage within the library environment.

3.3 System introduction

The system was presented to the client in the early stages of development to ensure a continuous flow of feedback. Stakeholders were able to interact with the system and provide the team with valuable insight on how to improve the way the dashboard should work and look to create more value for the librarians. Every week, during the stand-up sessions with the stakeholders, progress on the previous week's feedback was offered and new feedback was requested from them.

3.4 Usability testing

Usability testing was performed by the team members, stakeholders, and other students with no direct connection to the project. Each team member took a continuous approach to testing the features that the other members of the team implemented. This ensured that no edge cases or bugs were missed and that the final implementation was correct. Stakeholders were asked for input every week during the stand-up sessions to make sure that the project did not deviate from its original purpose and that if anything was misinterpreted, or could be improved, it was remediated as soon as possible. Students with no direct connection to the project were also asked to provide their feedback because of their fresh vision over the matter, mainly focusing on usability, user-friendliness, and intuitiveness.

3.5 Proposal presentation

At the end of the project, the system was presented to the management of BOZ BMS. This presentation showed the audience how the occupancy of the monitors can be visualized and what was the forecast for the following days. It also demonstrated the potential of expanding this application to other buildings such as ITC, or Horst, that have monitor spaces scattered throughout the buildings as well. A week before the final presentation, the team demonstrated the system in front of the supervisor's chair, along with other colleagues from the same department. This was an opportunity for the team to showcase the results and receive questions and final feedback points.

3.6 Results of meetings

Meeting with the supervisor and stakeholders resulted in overall useful feedback.

Weekly meetings with the client helped the team keep up with the timeline and fasten the progress. There were numerous brainstorming sessions to provide and obtain more detailed information about the requirements. Meetings became a fast way of communicating technical difficulties and finding solutions together, as well as foreseeing issues that might occur next. Specifically in the initial phases of the project, the team faced various problems with the collection and interpretation of sensor data. Meetings proved useful in providing a visual representation of the system, receiving feedback, and making adaptations according to the client's wishes.

Bi-weekly meetings were scheduled with the university's supervisor. It was essential for the university to know the level of technicality required for the design project and was also a way for the supervisor to provide useful feedback, and ensure the proper implementation of the system.

3.7 Conclusion

In Chapter 2 the problem of students and librarians struggling to track the monitors was addressed. A proposal was made that involved a management dashboard for the librarian composed of features such as 'integration of sensor data', 'trend analysis', 'user-friendly presentation', and 'maintenance and monitoring' to ensure the staff gets a full picture of the monitor area and take actions fast when unanticipated events such as defect monitor, unexpected overcrowded library, or other decisions which are inferred from the systems provided statistics. Besides that, a user dashboard will be provided for the students to check and monitor occupancy in real-time. Usability testing and weekly meetings with the librarians will occur to ensure a successful implementation of the proposal. From the initial meetings where the discussion in large held place, it helped shape the needed requirements and their importance which will be further discussed in the next chapter.

Chapter 4

Requirement specification and analysis

In this chapter, the process of specifying the requirements of the system is discussed. The requirement specification is guided by Agile approaches and practices, the iterative loops of development also occur in the requirement specification. The requirements need to be clearly stated when specifying them, therefore several techniques are used to identify the requirements.

In the second part, an analysis of the stated requirements is given. User requirements and system requirements are the two layers of requirements for the system that are being considered. These requirements are analyzed based on how well they satisfy the demands of the user and how quickly and accurately the containing features may be requested.

4.1 Agile project management approaches for requirement specification

Agile practices are used to manage this project. It keeps the development process and artifacts as light as possible, and makes the development process more seamless, by having more focused goals on development and collaboration, rather than on contractual negotiations between developers and clients [1]. Adding business value incrementally and iteratively proved to be useful for this project in meeting deadlines and expectations.

4.2 Requirement formulation

The process of requirement gathering started before the beginning of the designated period of the design project and also in the early weeks of the designated period. Due to the team choosing to reach out to the library for a project together with them, a few weeks before the beginning of the module were gained and used to the team's advantage to create a better understanding of the entire project while writing the project proposal. After several meetings, the team had an overview of the requirements and started a back-and-forth discussion with the library's staff to make sure that the vision and the road map were perfectly aligned.

The initial idea was the necessity of a way to manage and supervise the activity and occupancy around the monitor area of the library. This was the most needed feature and almost anything besides that was considered optional. The managing and supervising of the space had to be done on a dashboard available to both the student and the librarian.

The displayed information had to be as live as possible, without the personal information of the students being required to obtain or display it. Data prediction was considered a nice-to-have feature, along with other useful data points that could help the librarian in the decision-making process. Other features, like a monitor reservation system, were deemed undesirable for this project and, therefore not in the scope of it.

4.3 Requirement prioritization

Throughout this section, the **MoSCoW** (Must have, Should have, Could have, Won't have) prioritization model for user stories was kept in mind. This model helps with time-boxing certain tasks, where a fixed focus is on the most important requirements, and it also helps with understanding the "definition of done".

As a result of the design meetings the team had with the library's staff, multiple features surfaced, and for each feature, many requirements had to be created. Requirements help with creating a clear vision both for the team, as well as for the stakeholders, by putting in an overview of not only how each feature should look and function, but also what has priority. All requirements have been split into two categories, one for functional requirements, and one for non-functional requirements. User stories, on the other hand, have been categorized based on the Moscow prioritization model [5].

The multitude of requirements is split below into two categories: functional and non-functional. This way, a clear separation between how the system should work and how everything needs to happen can benefit all the parties involved in the design process.

4.4 Stakeholder requirements and system requirements

As noted in Chapter 2, the requirements are developed following the expectations of the stakeholders to expand upon their analysis. These prerequisites are referred to as the stakeholder requirements. System requirements are defined to determine the features the system should have to meet the demands made by the stakeholders.

4.5 Stakeholder requirements

4.5.1 Must have requirements

1. As a student, I want to be able to check the availability of a certain seat with a monitor on an online platform to avoid having to walk to the library to find out all of them are full.

2. As a student, I want it to be displayed if a monitor is unavailable due to it being defective.

3. As a library stakeholder, I want to ensure that the monitor area occupancy detection system complies with privacy regulations and respects user anonymity while still providing valuable usage insights.

4. As a library manager, I want to visualize patterns and trends in the monitor area. This will give librarians more insight particularly about the distribution of students in the library during specific intervals, for instance, exam periods, Christmas breaks, or peak times during the day.

5. As a student, I want to see all the seats with a monitor in the library and their availability to determine how busy that section of the library is and to avoid being left out without a seat.
6. As a student, I want to be able to check for how long a seat with a monitor has been free on an online platform to have a rough understanding of how busy that library area is.
7. As a library manager, I want to see the real-time occupancy of monitors on a matrix map. This information can be used to verify the physical usage of space in the library.
8. As a librarian, I should be able to observe the power usage of each monitor while on stand-by to make decisions towards sustainability if any monitors have to be replaced in the future.

4.5.2 Should have requirements

1. As a library manager, I want the monitor area occupancy detection system to be sensitive to diverse user needs, including accommodating individuals who may require longer breaks or have mobility challenges.
2. As a library IT administrator, I want to integrate the monitor area occupancy detection system with existing library management software so that I can streamline data collection and reporting processes.
3. As a library manager, I want to continuously monitor the health of sensors to ensure all sensors are functioning correctly, addressing any issues promptly to maintain data accuracy.

4.5.3 Could have requirements

1. As a library staff member, I want to receive alerts when there are prolonged periods of inactivity in the monitor area so that I can check for abandoned devices or address any technical issues.

4.5.4 Won't have requirements

1. As a student, I want to be able to see which monitor is reserved by me to know where I should be seated.
2. As a student, I want to receive notifications when my reserved time slot in the monitor area is about to expire so that I can either extend my session or vacate the spot for the next user.

4.6 System requirements

4.6.1 Functional Requirements

Each functional requirement encapsulates a feature from a technical point of view. It has the role of presenting how a certain aspect of the application should look like, in as much detail as needed for the developer and the stakeholder to be on the same page.

1. The system shall allow students to check the availability status of monitors.

2. The availability status displayed to students must accurately reflect whether a monitor seat is available or occupied.
3. The system should provide librarians a notification for inactivity.
4. The system must send alerts containing useful information like the inactivity duration and the exact monitor reference number to the librarian.
5. The system must provide alerts that have a spam prevention system.
6. The system should have the ability for librarians to mark a monitor as defective.
7. The visualizations for the trends should be intuitive for non-technical users.
8. The system shall implement robust access control mechanisms to restrict access to monitor area occupancy data to authorized personnel only, in accordance with privacy regulations.
9. The system shall provide mechanisms for users to access, modify, or delete their personal data as per privacy regulations.
10. The system shall ensure that all data collected, processed, and stored comply with relevant privacy regulations, including obtaining necessary consent from users where required.
11. The monitor area occupancy detection system shall be seamlessly integrated with the library's existing management software to facilitate data sharing and interoperability.
12. The integrated system shall automate the collection of monitor occupancy data from sensors and smart plugs, eliminating the need for manual data entry or manipulation.
13. The integrated system shall provide real-time reporting capabilities within the library management software, allowing administrators to monitor the occupancy and usage patterns in real time.
14. Library managers shall have the ability to customize occupancy thresholds based on factors such as user preferences, mobility challenges, and the need for longer breaks.
15. Threshold adjustments shall be configurable through the system's administrative interface, ensuring flexibility and adaptability to changing requirements.
16. The system shall incorporate user feedback mechanisms to gather input from individuals with diverse needs regarding the effectiveness of accommodations and services.
17. Analysis of historical data shall inform proactive maintenance strategies and optimization of sensor deployment to maximize reliability and minimize downtime.
18. Library managers should have the ability to observe the energy consumption of each monitor during stand-by.
19. Library managers should have the ability to observe the energy consumption of each monitor during live, normal usage.

4.6.2 Non-functional Requirements

Non-functional requirements refer to how the system operates, its performance, and its capabilities. Such requirements offer to all the parties involved certain benchmarks to aim for during the development process and in the end have the goal to create a product that has growth potential, increased usability and accessibility.

1. The users should be able to see a change in MazeMap within 5 minutes (the ingestion time of the sensors). [Performance - Response Time]
2. Both the management and user dashboards should respond within 5 seconds. [Performance - Response Time]
3. The system should be available throughout the library's opening hours. [Availability]
4. The system should be able to handle at least 500 users at the same time as that is the library's capacity. [Scalability]
5. Colorblind users should be able to navigate through the application without external help. [Accessibility]
6. A user should be able to use the dashboards for the first time within 3 minutes. [User-Friendliness]
7. The development team behind the application should work on different branches to facilitate easier code management. [Branch Management]
8. All merges to the main branch should be first approved to ensure code quality. [Code Review Process]
9. No external party should be able to tamper with the sensor data. [Security]

4.7 Conclusions

To sum up, the implementation of Agile project management techniques was crucial in guiding the project's requirement specification process. Agile approaches prioritized flexibility, cooperation, and incremental value delivery, which allowed for a smooth development process that was well-focused on satisfying the needs of stakeholders. A thorough grasp of the project's goals was made possible by the requirement formulation phase, which was started well in advance through proactive interaction with the library's stakeholders. This cleared the way for a distinct and cohesive vision. The team was able to prioritize requirements in an organized manner by utilizing the MoSCoW prioritization model, which also allowed them to concentrate on providing the most important features within the allocated time period. Clarity on both system functionality and operational characteristics was provided by the division of functional and non-functional requirements, which further expedited the development process addressed in the next section.

Chapter 5

Global and Architectural Design

In this chapter, the architectural and global design decisions are elucidated, and supported, and the system architecture is addressed at a high level. Additionally, a summary of the system is given, detailing its pages and general features.

5.1 Global Design Choices

The engineering of a system such as this system for the University of Twente's library is meant to manage the flow of people, scheduling, and for students to see the real-time occupancy of the monitors. In chapter 2, processes and procedures are identified and analyzed, unveiling directions in which the process can be enhanced.

5.1.1 Key Design Pillars

In the key design pillars, the foundational principles and priorities that guide the architectural design and development of the system are outlined. These pillars encapsulate the core values and objectives that shape the system's architecture, ensuring alignment with the system's goals and requirements. In a paper by Boehm, it is recommended to have a small set of basic principles in software engineering [3]. This project follows the key practices from this paper; for instance, using a **phased life-cycle plan**. It reflects on the importance of the project plan, the essence of the project overview, the phased milestone plan, project risk analysis, and a validation plan.

Prototyping and incremental development was practiced during this design project. 'Agile methodology' was used as mentioned in section 4.1. User stories were divided into requirements which were assigned to different members to develop. It helped with coding the critical portions of the software product as a means of user requirements determination, and with incremental development that takes into account the development risks.

Reliable software can only be created if proper validation is done. **Performing continuous validation** helped in getting the errors out early and building trust with the stakeholders. It involves continuously assessing the performance, functionality, and quality of a system throughout the life cycle. Continuous validation ensures that the system remains aligned with user expectations, business goals, and regulatory standards.

5.1.2 Redesigned Workflow

Through this system, the librarians and students have a different way to check the occupancy of the monitors. Smart library projects have provided functionalities such as alerting the students about a defect, and maintenance of monitors. Students can now make a more informed decision of whether to come to the library or not. Section 5.3 explains in more detail the interaction of the student and the system with the process of collecting the data from sensors.

5.2 Preliminary Design Choices

The first step in project management is to make wise initial design decisions. These decisions take the project team's knowledge and abilities into account. Rely on the knowledge and abilities that have gone into creating the programming environment, taking into account the programming languages, frameworks, and libraries that have been employed.

5.2.1 Programming Language

Python is well-suited for developing the smart-library system. It is easy to learn and use because of its simplicity and readability, making it accessible to both new and experienced programmers. It is advantageous in a project where in the future it might need to communicate with other UT software that has the same stack as well. Python provides a wide range of libraries and frameworks, more in detail in the next subsection. It is also versatile for integration with diverse technologies and data sources, aligning with the project's requirement to accommodate different types of sensors and data. Python's interoperability capabilities enable seamless communication and data exchange between different components of the system, enhancing overall functionality and user experience.

5.2.2 Frameworks and Libraries

For this project, **Flask** was used, which is a Python framework. It is lightweight, independent, flexible, and compatible with machine-learning libraries. The team chose Flask because it is suitable for small projects and it is easily scalable.

Initially, multiple API calls were made in order to access data. This resulted in an increased latency in showing the real-time occupancy of the monitors. To tackle this, the decision to have a database was made. **Flask SQLAlchemy** was implemented to store the sensor data and the login details of the admin. This saved development time and increased the maintainability of the software. Flask SQLAlchemy improved the loading speeds for graphs and occupancy information for the 'Management-dashboard'.

There are different types of models used in prediction and analysis for the power/occupancy of monitors. Before the selection of models, cleaning the data was required. **Pandas** Python library was an integral part of the easy manipulation of sensor data. Also, its statistical tools were useful in analysis. During the initial phases of the project, the team was required to visualize the data for a better understanding of different variables and the structure of sensors. **Matplotlib** helped in making fast graphs to recognize trends and patterns. These methods helped in defining strategies for making a prediction model for occupancy and power. Furthermore, **ARIMA** was initially used for forecasting which was later switched with **XGBoostRegressor** due to better accuracy results.

A simple approach was chosen for the front-end implementation, using **HTML**, **CSS**, and **JavaScript** as the building blocks. A responsive and aesthetically pleasing user experience that meets the various needs of the consumers by utilizing these fundamental technologies was developed as a result of this.

To improve the system's functionality and visualization capabilities, two potent JavaScript libraries in addition to the essential front-end technologies were included. First, the **Chart.js** library was used to make interesting "donut" type graphs and dynamic widgets. With the extensive toolkit and customization options that Chart.js offered, presenting complex data in a clear and understandable manner was facilitated.

Moreover, **Plotly.js** tool was used to produce predictive insights and support well-informed decision-making.

5.2.3 Architectural Design choices

A collection of key decisions concerning the design of the software of a system is considered as software architecture. An architectural design helps in codifying design choices. It has two advantages, firstly making an explicit bridge between design and rationale. Secondly, architectural design brings new perspectives and deepens the understanding of the software. [4]

In the architectural design of this project, several key considerations were taken into account to ensure the robustness, scalability, and maintainability of the system. The system follows a **client-server architecture** where the front-end (client) communicates with the back-end (server) to retrieve data and perform operations. This pattern allows for the separation of concerns, making it easier to maintain and scale each component independently. **RESTful API** to facilitate communication between the front-end and back-end. This style enables a standardized and stateless communication protocol, allowing for efficient data exchange between client and server. By following REST principles, the system promotes interoperability and flexibility, making it easier to integrate with other systems in the future. The system may employ a **micro services architecture**, where functionality is divided into smaller, independent services that can be developed, deployed, and scaled independently. This architectural approach promotes modularity, flexibility, and fault isolation, allowing for easier maintenance and scalability of the system as a whole. Finally, **version control and documentation** tools like GitLab and maintaining comprehensive documentation ensure transparency, collaboration, and traceability throughout the development cycle. Well-documented code, architecture diagrams, and API documentation facilitate the onboarding of new team members and promote knowledge sharing within the development team.

5.3 System description

5.3.1 Class diagram

The class diagram can be found in figure 5.1. In the library monitoring system, the class diagram is an important tool for understanding the different components and their interactions. The system includes several modules such as **Monitor data analysis** and **Smart Plug data analysis**, which call APIs to access sensor data and perform preprocessing actions. The **Monitor Area Predictor** combines data from both sensors into one data frame, while the **Data processor** uses this information to provide more reliable data on monitor occupancy and perform time-series forecasting and analysis using XGBRegressor. The **Graph** module is responsible for plotting the graph of monitor occupancy over time. The **Monitor display** returns the status of monitors, indicating whether they are available, occupied, or defective. **Student alert** is a module that creates alerts relevant to students, while **Monitor defect alert** alerts librarians if there is suspicious activity such as not receiving monitor information. The **Admin** module has predefined login credentials and can view graphs, monitor information, create alerts for students, and monitor suspicious activity. Finally, **students** can view alerts created by the staff and monitor occupancy of the library monitors. By working together, these components ensure the smooth functioning of the library monitoring system.

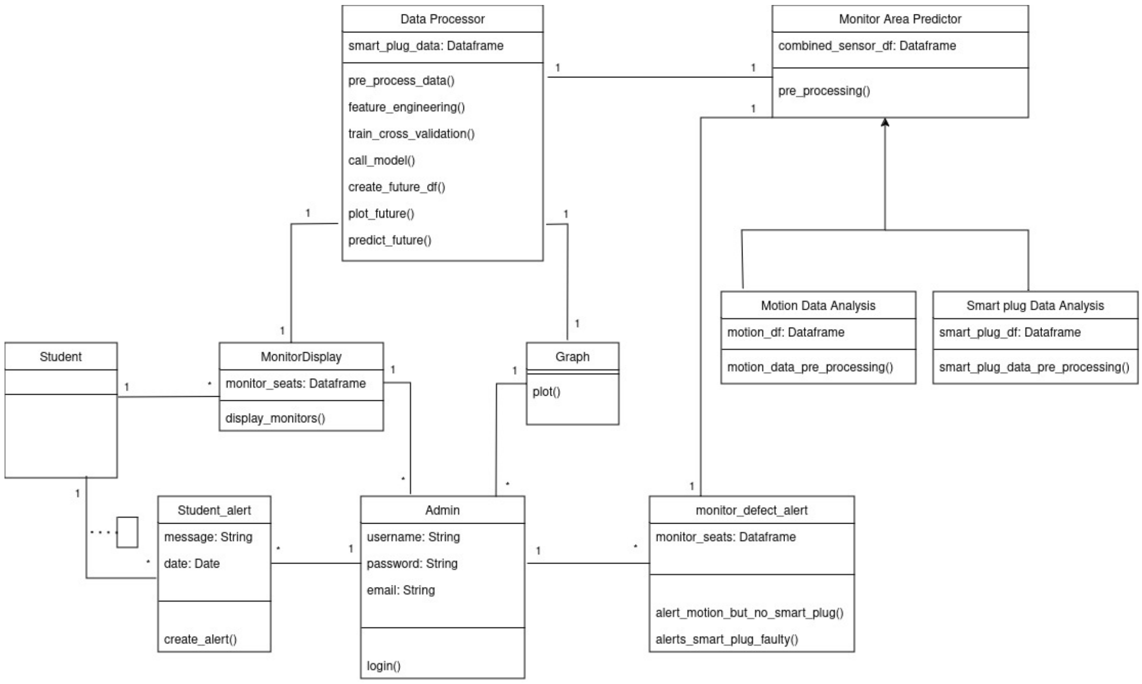

FIGURE 5.1: Library Class Diagram

5.3.2 Flow diagram

In Figure 5.2 the flow diagram of the overall system is presented. This can be separated into four swim lanes: Data Collection, Database Management, Dashboard Management, and Library Staff. The "Data Collection" phase is where the two main sources of data are obtained: the smart plugs and the motion sensors. The two sensors constantly measure data and forward the information to the library's database. In the "Database Management" swim lane everything is being collected, therefore only the data relevant to this project is then stored in an internal database. This not only helps with maneuvering large amounts of data easier, but also with historical aspects needed for the data analysis presented on the dashboard. The information in the internal database is later used in the third swim lane, the "Dashboard Management", to create data visualizations that are useful to the librarian. After processing, this data reaches the final swim lane of the flow diagram, the "Library's Staff". In this step, the data is presented in the final form to the librarian, who can also interact with some aspects of the data.

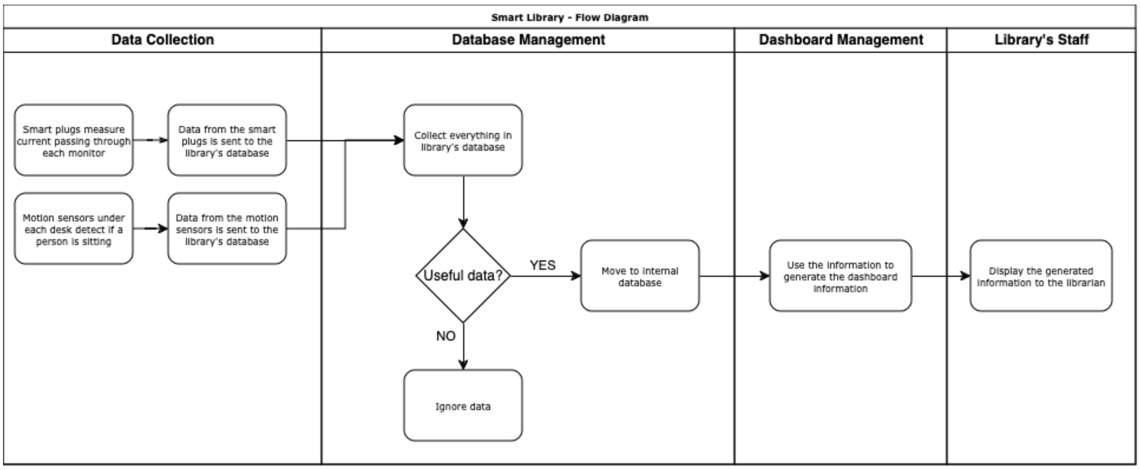

FIGURE 5.2: Flow Diagram

5.3.3 Database diagram

Figure 5.3 shows the database schema for this project. During the initial phases of the project, we did not have a database because the LISA department already had a db (Influx db) to which our system was doing API calls. The teams realized that it was not the most efficient way of using the data as doing multiple API calls for different features takes more time. After this, we created a really simple database that stores the sensor data and login credentials for the user. It shows independent entities such as users and student alerts. The statistics, smart plug, and motion sensor table are connected to the sensor table which is a 'has' relation. This structured approach allows for efficient data management and retrieval in applications that rely on this database.

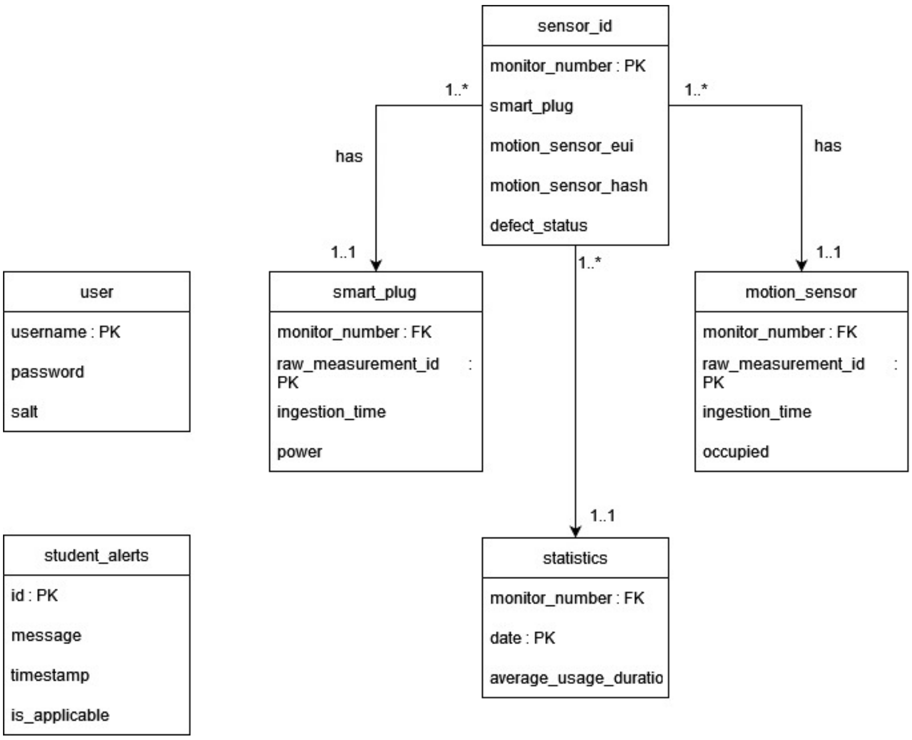

FIGURE 5.3: DB schema

5.3.4 Use case diagrams

In diagram 5.4 the intended use of the Mazemap interface can be seen. Students and librarians can use Mazemap to view the real-time monitor occupancy. The overview also shows which monitors are defective If they wish to see the information about a specific seat they can select it and the system will show how long it has been free for. The relationship between "View Monitor Occupancy" and "View Overall Monitor Seat Placement" is marked as "include" since the seat placement view automatically shows the occupancy.

The second use case diagram 5.5 shows how librarians can use the web application developed specifically for them. Library employees can use it to view information about the monitor's health in addition to other trends derived from the sensor data, such as the average study duration, power consumption, and a comparison between the overall library occupancy, versus monitor area occupancy. They also have the option to view the monitor places on a map as well as add alerts which show on the student view page.

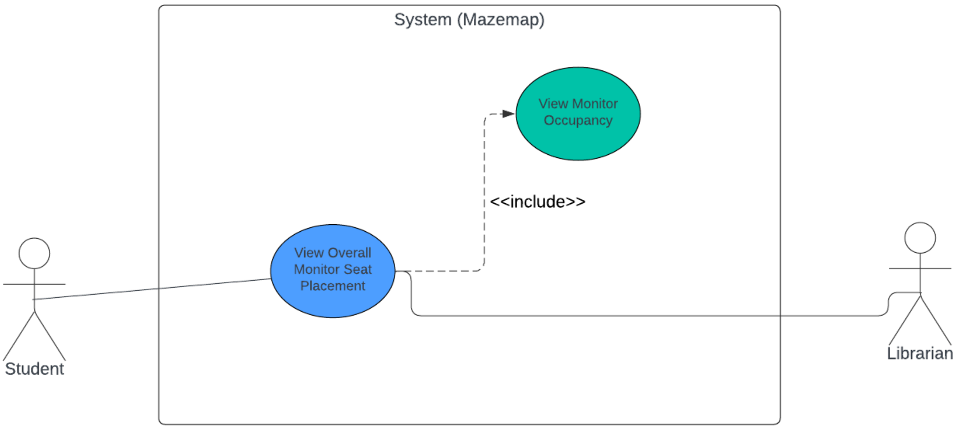

FIGURE 5.4: Use case diagram - Mazemap

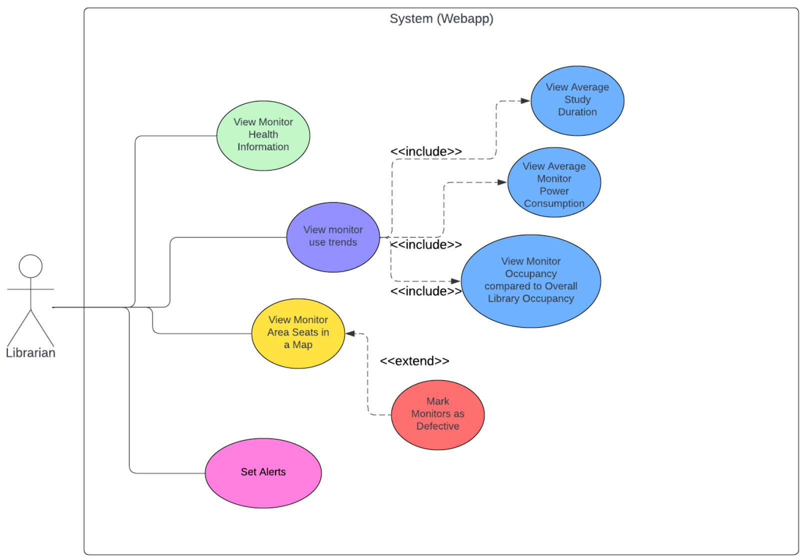

FIGURE 5.5: Use case diagram - Webapp

5.4 System Overview

This section aims to justify the design choices made by the team. Design choices were presented to the client weekly with new and remodeled interface features based on their feedback. These components serve to aid the user experience and are designed to enhance the ease of use. In the following paragraphs, these components are elaborated. There are two new dashboards: The librarian admin dashboard and the student dashboard. Each unit will be explored in detail, explaining the design of every existing feature.

5.4.1 Librarian admin dashboard

Login page

The purpose of the login page is for the library staff to authenticate themselves by using the predefined credentials created for them by the team. This is a simple login system that has the role of preventing students from accessing the information on the dashboard. This information is only relevant and meant for the staff, therefore, the login page will always request a username and a password before the user can see the dashboard.

Before accessing the library admin dashboard, they are presented with the login. The initial design for the login page is provided in figure 5.6. The library staff will be given the necessary authentication credentials. This design choice ensures no one except the staff can view the admin page. The team also opted to provide them with authentication credentials rather than allowing them to create an account on the dashboard to prevent unnecessary verification requirements.

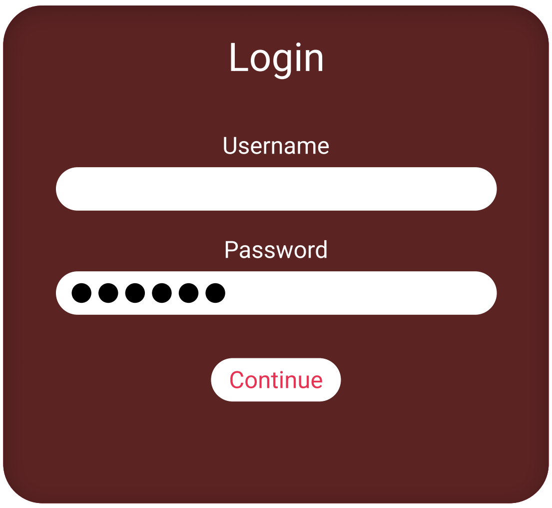

FIGURE 5.6: Initial design for login page.

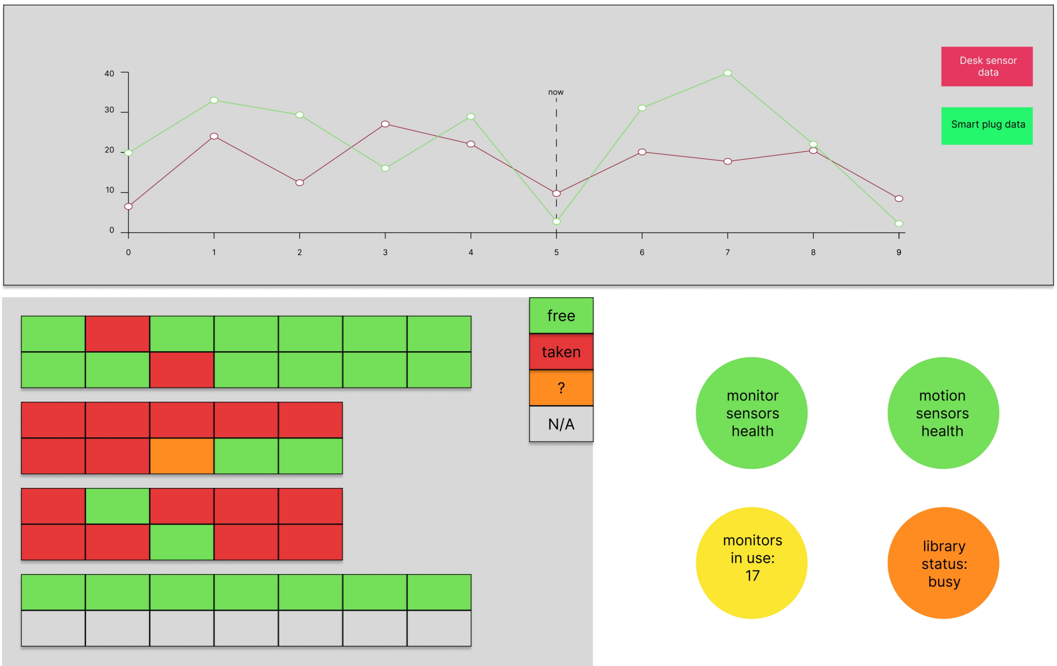

FIGURE 5.7: Initial dashboard design

Library admin view

The library admin view, also presented throughout the paper as "library admin dashboard", provides admins with various information about the library, such as: a close to real-time representation of the monitoring seating area occupancy, occupancy forecasting, list of detected defect monitors, monitors' energy usage, list of malfunctioning sensors, widgets for existing alerts, and widgets for creating alerts for students. The purpose of this dashboard is to ensure the library staff is fully aware of the monitor area surroundings to perform necessary actions. Given that the employees do not possess technical expertise, extra care is provided regarding how data is presented.

For the general feel of the platform, while the aim was not to make it tech-savvy, user-friendliness was also not the priority. This is because the platform is pretty simplistic in features and the library staff will be trained to use it. An instruction manual is provided.

The team's goal was to create a simple design that presented all the features on a single page. This helps the staff to easily locate the main functionalities without getting lost in multiple navigators. The layout is structured in a grid form where the more important features take a wider space as shown in figure 5.7.

Monitor area occupancy

The aim for the monitor seat UI was to replicate how they are positioned at the library, making it much easier for the librarian to spot the appropriate monitor. For displaying the monitor status, standard colour codes were used, such as green for available, and red for occupied. Orange is used to indicate suspicious monitor activity.

Finding out if a monitor was occupied or not proved to be a difficult challenge. Since a monitor's occupancy depends on not just the current data, but the way the data has changed, we had to figure out what algorithm would best represent our interpretation of the data. We did this by manually going through the graphs of smart plug and motion sensor data and labeling when someone got up or sat down. Then we used a sliding window protocol to go through 'moments' in time as a group, to detect large changes, which are big enough to warrant a recalculation of whether something is occupied or not.

Forecast

The forecasting module of the project provides power output for the next 3 days. Power output can be correlated to two factors the occupancy and energy consumption of the monitor area. The number of people occupying a certain area can impact the energy consumption of that area. Higher occupancy levels typically result in increased usage of electronic devices leading to higher overall power consumption. Understanding the occupancy and energy consumption is seen as essential for our stakeholders 'sustainability goals'. By accurately forecasting energy demand, librarians can optimize resource allocation and reduce waste. It can help them in infrastructure planning. Librarians can schedule maintenance tasks such as repairs, equipment inspections, or upgrades by identifying periods of low energy demand. They can make informed decisions about upgrading or expanding the energy system within the library. An initial implementation of the forecasting system can be see in figure 5.7.

Monitor alert system

There were two options for alerting staff about defective monitors: via email or directly through the platform. The team chose the second option to prevent potential spam in the librarian's inbox.

There are three possible defect states: number of faulty motion sensors, number of faulty smart plugs, and number of suspiciously used monitors. All defect states are highlighted in orange, and a summary circular progress bar widget is displayed for each of them.

A sensor is deemed defective when no data has been received from it in the past 24 hours. A suspiciously used monitor is highlighted when a student sits at the desk without having the laptop connected to the monitor. This is done by using a combination of data from the motion sensor and the smart plug. This is undesired because the monitor is unused, and could otherwise be used by another student that needs it.

Setting a monitor as defective

The aim of this feature is to provide librarians with the ability to set a monitor as defective. Figure 5.8 a simple and clear implementation for this feature. It was quite straightforward to have a pop-up on every monitor to make the monitor defective.

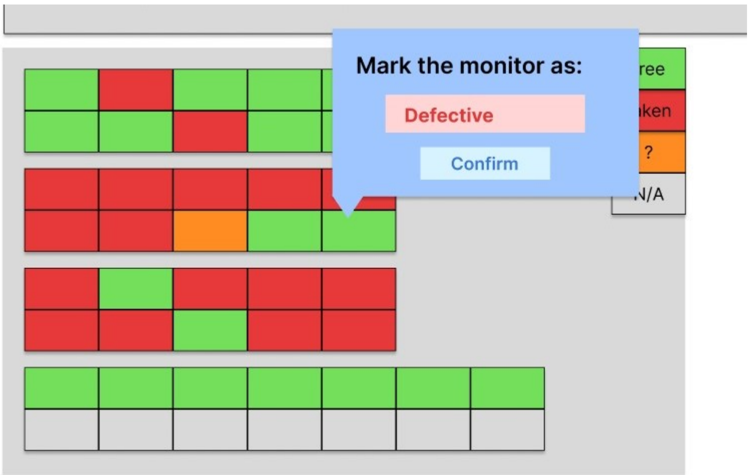

FIGURE 5.8: Initial design for defective monitor

Student alert system

The alert system is a widget that allows staff to create alerts for students by clicking on it. To prevent human errors, such as unintentionally pressing the enter key too early or making spelling mistakes, the staff can view the alerts they have created and modify them if necessary. Figure 5.9 is an initial design for the alert system. A limit of three alerts per staff member was set to prevent the student page from becoming overloaded with alert pop-ups and to avoid the need for creating too many alerts. When deleting each alert, they are first presented with a confirmation prompt to prevent accidental deletions.

FIGURE 5.9: Initial design of alerts.

5.4.2 Student dashboard

A separate web page is meant for students and serves two purposes: showcasing close to real-time monitor occupancy representation and alerts meant for students. No authentication mechanism is required to view this page which makes it freely accessible by any student, and no personal information is required or stored about the users. When you access the dashboard, alerts will pop up on the screen. You can close them by clicking on the "x" button. If you only want to see a summary of seat occupancy, you can view the maze map displaying the monitors on the web page. However, the Mazemap is also clickable and will redirect you to the actual Mazemap if you need to interact with it. During the initial phases of the project, these features were supposed to be implemented into Mazemap directly. That's why we did not have an initial design for this feature. Later, we found out that Mazemap could not support such features, so we implemented a basic web page shown in subsection 7.1.2.

Mazemap view

In the Mazemap testing environment, an overview of the monitor occupancy can be found. Due to Mazemap's limited features, currently, only the monitor availability is indicated in green or red.

5.5 Additional sources

Besides the described, a manual will be provided to the librarian with step-by-step instructions on how to use the platform. This includes how to log in and how to use each of the available features. Code is also properly documented using agreed-upon Python and Flask coding conventions such that other developers can take over and add more features.

5.6 Conclusion

In this chapter, we delved into the global and architectural design aspects of the Library Management System of the University of Twente Library. We began by discussing the most important design pillars, which are the basic principles of system architecture design and development. These pillars, including phased life cycle design, prototyping and incremental development, and continuous validation, ensure alignment with project goals and requirements. In addition, we explore initial design options, such as choosing Python as a programming language. and Flask as a framework, with various libraries such as Pandas, Matplotlib, and Chart.js contributing to the functionality and visualization capabilities of the system.

Architectural design decisions were also explored to ensure reliability, scalability, and maintainability, including client-server architecture, RESTful API, and microservices architecture. We also provided a detailed system overview that highlights library management capabilities, panel, and student panel. The Librarian Dashboard provides insight into monitoring area usage, forecasting, error tracking, creation of alerts for students, and in-place alerting systems for library staff needs. On the other hand, the Student Control Panel provides students with a real-time display of monitoring usage and alerts, which improves their user experience.

Finally, the design choices during the development process aim to create a user-friendly, efficient, and reliable library monitoring system that meets the requirements of both library staff and students. By following well-established design principles and using appropriate techniques, the system is well-equipped to serve its purpose effectively. Now that the features have been discussed at great length, the next chapter aims to discuss important testing strategies imposed throughout the development phase to ensure the reliability of the product.

Chapter 6

Test Strategy

In this chapter, the test plan and the test results are provided. The test approach is explained, different functionalities that are tested are indicated, test criteria are provided, the schedule is discussed and the risk and contingencies are indicated.

6.1 Test Plan

6.1.1 Approach

The testing strategy is important in assessing the correct interaction between all the modules of the system. This involves tests for the flow of data from the sensors to the database, as well as from the database to the dashboard. This ensures the correct processing and presentation of the data towards the user. In order to test the system, it is tested according to four strategies. These strategies are a) Unit testing, b) Integration testing, c) Manual testing d) Usability testing. The purpose of the test plans is to isolate and break down the system into small components that can be tested individually to ensure correct functionality. Such tests focus mainly on three broad categories, the sensor monitoring system, the database management, and the dashboard interface.

6.1.2 Software risk issues

There are several critical areas of the software to be tested. Firstly, is the showing of the occupancy of monitors at both the librarian dashboard and on Mazemap. Secondly, the statistics widgets should show correct values for suspicious monitors, free monitors and defect sensors. Thirdly, significant effort was dedicated to enhancing the user interface and experience (UI/UX) of the system. This included addressing issues related to the loading and reliability of external libraries used for graphing. At times, certain libraries experienced downtime, necessitating prompt resolution to ensure uninterrupted functionality.

6.1.3 Functionalities to be tested

This section lists what should be tested from a user's perspective. Functionality requirements have already been analyzed in chapter 4. The level of risk for each functionality is rated using High (H) or Medium (M) based on how important it is to test the functionality. Refer to the table 6.1 for these risk indications.

Furthermore, the front-end of the system is tested based on its usability for the end-users.

Functionalities to be tested	Level of risk
Showing occupancy of monitors on the librarian dashboard.	High
Showing the number of defect sensors accuracy.	High
Showing the number of suspicious monitors.	High
Showing the base values for every monitor.	High
Ability to change a monitor status to 'defect'.	High
Ability to integrate the people counter.	Medium
Showing the prediction for power production.	Medium
Ability of the librarian to make alerts through the system.	High
Login/Logout system for the librarian.	High
Showing occupancy of monitors on the Mazemap.	High
Showing alerts to students.	High

TABLE 6.1: List of functionalities to be tested and their corresponding level of risk.

6.1.4 Item pass/fail criteria

To release the product it is essential to have test cases for all the medium and high-priority cases from table 6.1. High-priority cases were tested more extensively and unit tests were made where possible.

6.1.5 Schedule

An agile methodology was used in this project. It means that requirements were changing during the development of the system. Most of the units were made during the end phases of the project, so we don't have to adjust the tests according to any requirement changes. Integration testing had a different approach, it was done any time a feature was added through the development process. Manual testing was the most important test for this project. It required us to do manual testing during the entirety of the development process. Unit testing was performed to ensure the correct outcome was being showcased for possible actions. Finally, usability testing was used towards the end of the project as it is primarily used to tweak user interfaces.

6.1.6 Risks and contingencies

Risk analysis is a critical component in the development and execution of projects for several reasons. By identifying potential threats and uncertainties early on, a plan to mitigate the issues in the unfortunate chance of encountering them can be created. The team is going to address the potential risks, their cause, their importance, and a mitigation risk to abide by if needed. Some of these risks were identified in the early stages of the development, while other risks emerged during the development process.

Risk: The risk of the team not being aligned in their vision.
Cause: Not everyone getting a grasp of what the main requirements were. This could result in unnecessary or incorrect implementation.
Level: High. Due to the team consisting of 5 members, with different schedules, it was proven hard to schedule meetings where the complete team was present. This leads to gaps in the information that reflects on the project's road-map. Every interpretation of a topic is different, especially when information is missing, therefore, there is a high risk of this threat to occur.

Mitigation: Mitigating this risk was obtained with precise communication and planning. Information was written down during meetings and passed forward to the team. The team was able to maintain an efficient line of communication with all the stakeholders involved, as well as with the supervisor and the product owner. Digitally, every task was recorded with Trello, where deadlines, levels of priority, and people were carefully assigned to improve productivity. All the user stories and their corresponding set of requirements were created and monitored inside the same Trello board.

Risk: The risk of emerging scope creeps.
Cause: During the development phase it was noticed that requirements started to grow in directions which were not intended initially. Some requirements were left ambiguous which led to the free interpretation of the assigned developer. Such scope creeps went overlooked during the development stage and surfaced only in the late stages which led to wasted resources and a shift in focus.
Level: Medium. Due to the developing dynamic used throughout the sprints. The dynamic consists of each developer focusing on a certain feature. This was due to the independence between the tasks that made it difficult for multiple developers to work on the same feature at the same time.
Mitigation: Mitigating this risk was done by continuously querying about the status of each feature. This helped in detecting such scope creeps in the early stages therefore saving resources from being wasted on undesired features.

Risk: Unbalanced skill levels inside the team
Cause: Each member of the team has various passions and expertise levels in certain domains. This is due to the previous courses each member took during their electives and minors modules. This can resemble each member's ability to perform a certain task or requirement in a timely manner.
Level: High. Since this project is heavily relying on data science algorithms, a good knowledge foundation in this domain is important. Since courses that cover such topics were only taken by a few members of the team, these members were the ones to tackle advanced challenges. The remaining challenges were distributed to the rest of the team to create an even working field.
Mitigation: The imbalance was mitigated by assigning tasks appropriately to properly make use of all the team's resources. The team members know each other and each member's ability which made it easier to assign tasks based on their expertise. By doing this, less time was wasted on tasks, which in the end meant less stress during the sprints.

Risk: Extensive product future implementations.
Cause: The product has great potential for future implementations, but due to the limited time this project receives, some of the aspects were left out or marked as "could have" or "won't have".
Level: Medium. Since this project is done for the library of the University of Twente, the stakeholders have a clear idea of what the product looks like. This implies both what they know is needed but also what is not needed for this project to be complete. Each year they have various projects made by students, therefore, each feature that this project or other project does not cover, but in the future will be nice for the library to have, and can be covered by future groups of students.
Mitigation: Having a clear road map and goals, as well as a shared vision of how the final product must look like is what kept the team on track. The team aimed to finish

all requirements before week 8 of the project such that any additional features could be implemented if required by the stakeholders, product owner, or supervisor.

Risk: Unsatisfied students.
Cause: While the product owner, the stakeholders, and the supervisors might be satisfied with the product, there are always the students that need to be taken in consideration as well. Since this is also a product made for them, not just for the librarians, their opinion matters and if at least some of them do not get consulted about it, the entire point of the project might be missed after deployment.
Level: Low. Since the feature used by the students is minimal, and the librarian uses the main functionality of the application, the risk of having disappointed students is low but never non-existent.
Mitigation: Interacting with students via surveys and direct discussions, as well as discussions with the product owner is necessary to make sure the project is correctly developed. A useful aspect is that all the team members are often users of the library's study spaces and have a clear idea of how a product like this can be perceived from the user's side.

6.1.7 Approvals

Determining the completion of the testing process requires collective evaluation by the entire project team. The system must align with the end-users requirements to ensure functionality as per their needs. Hence, final approval of the system should also involve the end-users.

6.2 Test results

6.2.1 Integration tests

The occupancy of the monitors was implemented on Mazemap. It is a service that was provided to our project by LISA. Therefore, there was limited testing for this functionality during the development and usability phases. No problems have arisen during the integration testing as expected.

There was, however, another issue that arose during this phase. Due to the nature of our database and our code being quite optimized for the server, our local testing environment ended up running quite slowly. This made operations like refreshing slow on the local side, however, we ensured that this wouldn't be an issue once it was on the server itself. This has to do with the way that we cache our data and make calculations, so on the live server, it's possible that for the first few minutes of it running, when it's making a lot of calculations, it would take up to a minute to load, but after that, it will go back to its original speed.

6.2.2 Manual tests

Manual testing was the most crucial part of testing. The team manually checked if the sensor IDs provided by the library were correct or not. Inspection of sensors under the desks was done by crawling under the desks to assure proper functioning of the sensors. Our team sat down in the library for hours checking the monitors and our dashboard for the accuracy of the system. In this process, we found a lot of edge cases. For example, a monitor draws significantly more power if a laptop is plugged in via the monitor's USB-C and the laptop is charging. Another example was related to the brightness of the monitors.

Someone was using the monitor at 10 percent brightness, it was observed that the new monitors draw little to no power. It was interesting to find out these cases.

It was also made in front-end by ensuring everything looks and works as it should. For example, if a library staff creates multiple alerts the alerts have to be displayed nicely, be readable, and removable. If the librarian marks a monitor as defective, it should change colors, and other examples follow this way.

6.2.3 Unit tests

While manual testing was useful in many cases in ensuring our product's reliability, it used live data. However, this does not ensure our product works on uncommon occurrences such as defective motion sensors or when everyone is sitting on the monitor without utilizing it. This is where unit testing came into play. Methods that are responsible for displaying the 4 alerts in Figure 7.2 were tested.

Occupied seats: Simple unit tests verifying whether the number of occupied seats in a dummy data frame matched the one returned by the method responsible for occupancy.

Defect motion and smart plug sensors: When there is no sensor data after more than 24 hours it is labeled as 'defective', thus testing was done by adding and removing sensory data entries and checking whether the method detected this. Edge cases such as sensor data exactly on the 24 hour mark was also considered.

Suspiciously used monitors: The condition happens when the motion sensor detects the seat as occupied and power is below the required threshold. Thus, different unit tests were created such as whether the program correctly labels the monitor as suspicious if the condition holds and vice versa.

Overall we were able to spot some bugs while crafting our unit tests. One bug was that when the query for the sensor data during the 24 hour mark is empty, the method would crash as it would try to fetch attribute data even if list was empty. Another issue was that it was not working for the exact 24 hour mark. After identifying this problem, steps were taken to ensure that similar errors were avoided in the implementation of other methods.

Examples of some unit tests can be seen in 6.1.

```
aiman *
def test_is_monitor_occupied_by_power_below_cutoff(self):
    # Tests that the monitor is considered free when the power reading is below the cutoff value.
    moment = datetime.now()
    motion_data = [MockData( power: None, moment - timedelta(minutes=1))]
    interpolated_data = [MockData(power=8, ingestion_time=moment - timedelta(seconds=30))]
    plug_cutoff = 10
    result = is_monitor_occupied(moment, motion_data, interpolated_data, plug_cutoff)
    self.assertFalse(result)

aiman
def test_calculate_occupied_monitors(self):
    # Mocking fetched_data with a sample data set
    fetched_data = {
        'monitor1': {'recorded': 'Mon, 01 Jan 2020 00:00:00 GMT', 'value': 1},
        'monitor2': {'recorded': 'Mon, 01 Jan 2020 00:00:00 GMT', 'value': 0},
        'monitor3': {'recorded': 'Mon, 01 Jan 2020 00:00:00 GMT', 'value': 1},
    }
    expected_available_monitors = 46  # 48 total - 2 occupied
    result = calculate_occupied_monitors(fetched_data)
    self.assertEqual(result, expected_available_monitors, msg: "The available monitor count is incorrect.")
```

FIGURE 6.1: Unit tests for monitor occupancy

6.2.4 Usability tests

Usability testing is an important but overlooked step in many projects. We made sure to prioritize this as well were developing our application, to ensure that the largest number of people possible could use the application. Luckily, the majority of our application's features are to be used by the librarians, and we had weekly standups with the library. The great thing about this continuous feedback was that every week was essentially a usability test since it allowed us to show our main users, those with and those without technical knowledge, the development process and ensure that all features made sense to them and worked as they envisioned. In addition to this, we wanted to make sure that the students would also be able to use our application without issues. To this extent, we had various students try out the Mazemap app, to see if they'd be able to find the monitor area on it. We did this on both the computer and the mobile application, the only difference being, that we're unable to access the staging environment from the mobile version, so we only made them find the monitor area, but the area they found would have been uncolored.

6.2.5 Conclusion

Testing was crucial during our project weeks and various test strategies were used to ensure the meeting of deadlines for each sprint. Firstly, a test plan was formed during the planning phase where the main functionalities were rated with a risk level from 'Low' to 'High'. The functionalities rated 'high' were more extensively tested during the other phases. Possible risk analyses that were anticipated before the development phase were made and the mitigations that were performed during the mitigation phase. Lastly, various tests such as 'unit tests', 'integration tests', and 'manual tests' were performed during the development phase to catch unanticipated issues, and ensure the reliability of our product. For functionalities where unit testing was easily possible, such as the four alert features, we ensured full coverage of the methods. For others, such as API calls, prediction graphs, and UI changes, manual tests were performed. Lastly, usability testing was performed throughout the product by weekly meetings with the librarians showcasing and updating our visual designs and also having some students test our product.

Chapter 7

Results

This section will offer a summary of the entire design project, while highlighting the results and findings.

7.1 Outcome of Project

As a final product, the project covers and solves the initial problem statement that was offered to the team in the beginning of the design project. The outcome is a favourable one and satisfies not only the initial requirements of the stakeholders but also any requests they had throughout the development period. In its final shape, the project creates more utility to the librarians than in the initial phase, by adding extra data points that were deemed important in the later phases of the development process. Some examples of such data points are the sustainability aspects, study duration, and alert systems.

The final interface and functionalities were presented in previous sections and during the demo. For a detailed explanation of the functionalities please refer to section 5.4. An overview can be seen below:

7.1.1 Staff page

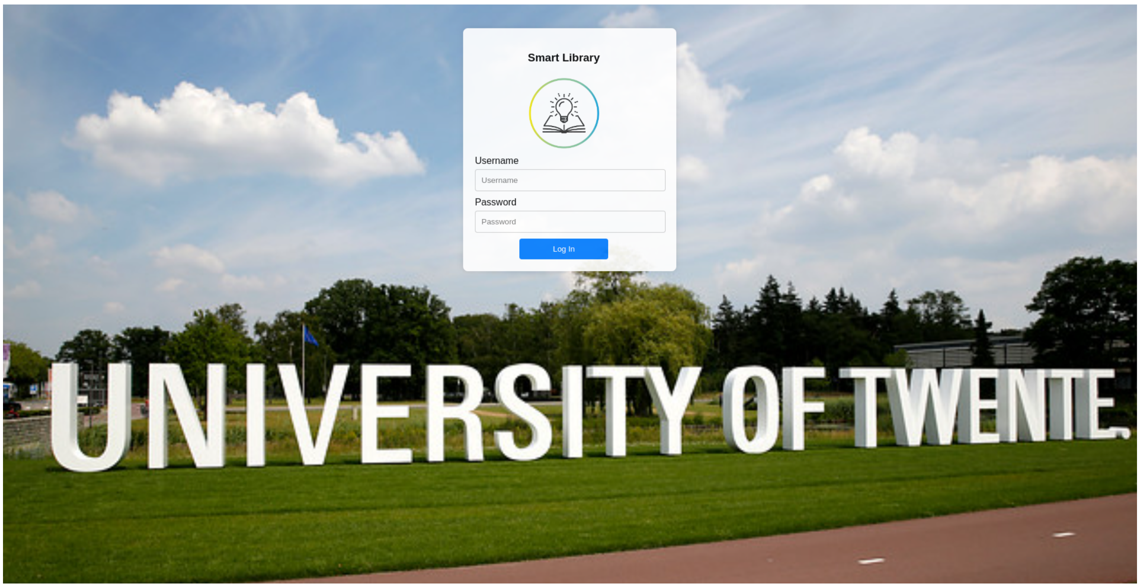

FIGURE 7.1: Librarian login page

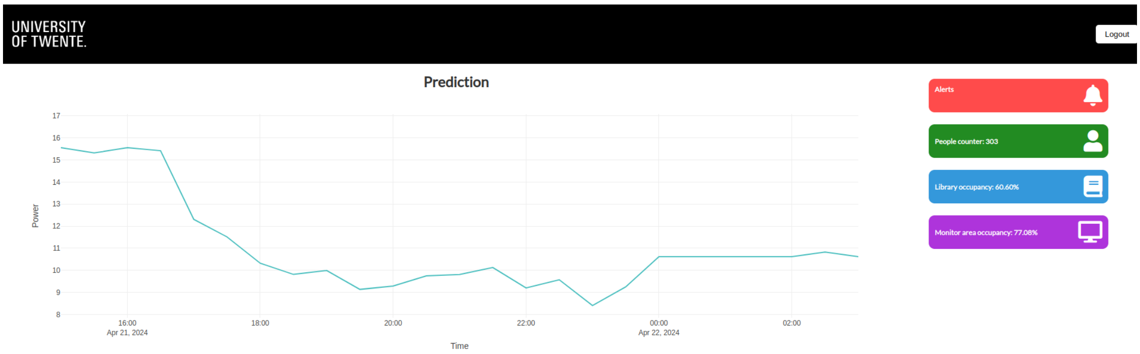

FIGURE 7.2: Librarian staff webpage 1

FIGURE 7.3: Librarian staff webpage 2

7.1.2 Student page

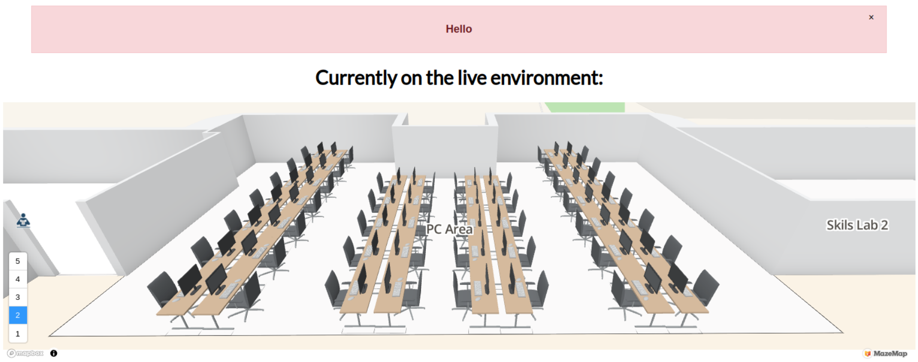

FIGURE 7.4: Student webpage 1

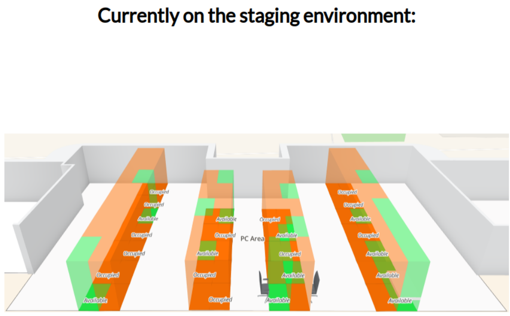

FIGURE 7.5: Student webpage 2

7.2 Future Development

The project represents a solid base for future projects to be built on top of. The monitors area inside the library will still be one of the most looked after areas to study inside the library and that opens more opportunities for other projects to experiment with. As future development features, some of the aspects that the team discussed with the stakeholders but was deemed undesired at the current stage is a booking system for the monitor area, as well as a better integration with Mazemap. The reasons for postponing these features are the plans for creating a new booking system at a university level, and the limitations that Mazemap has in terms of displaying information on their platform. There were a few other nice to-haves, such as a colorblind-friendly pallet, which we were also unable to complete because of Mazemap not allowing us to change colours. Finally, one other thing that was very briefly talked about was the possible integration of this system into other parts of the campus, for example, Ravellijn also has some of the desk sense sensors.

7.3 Timeline of Project Development

The timeline of the project ended up being a standard one. It started with extensive discussions with the stakeholders, followed by a requirement elicitation phase and multiple design sessions. The majority of the time was spent on the development process, with this report being developed in parallel. The final weeks were spent tidying up everything, testing, and creating the poster and the final presentations.

7.4 Reflection on the process

As a team, a reflection on the entire process was done on multiple levels. In terms of timeline, the team did great in distributing the workload symmetrically across the entire period. The entire team was excited about the utility of the project and felt like the final project would ultimately be useful in their student lives as well. At a motivation level, this improved the desire of the team to work towards the final goal of this project. At an outcome level, as a collective, the team and the stakeholders are happy with the final result, which is the most important aspect.

7.5 Personal Contribution

The team distributed the workload evenly towards its members according to their expertise and their passion for a certain functionality. Therefore, all team members were able to tackle issues both in the front-end part of the project, as well as in the back-end part and the final report. The project benefited in the early stages from brainstorming sessions that helped both the team and its stakeholders to express their ideas and opinions about how the project should look in its final form.

7.6 Conclusion

In conclusion, the project is a success. Everything of major importance was fulfilled and every party involved is happy with the final results. The initial problem statement now has a solution with great future and scalability potential, and both the students and the librarians have a useful dashboard providing them with the necessary information about the monitor area inside the library.

Chapter 8

Evaluation

In this chapter, an overall evaluation of the project is provided by means of elaborating on its planning, identifying the different responsibilities within the project team, evaluating the project team, and discussing the final result. Finally, this chapter concludes this report by stating the conclusion of this project.

8.1 Planning

In the following sections, a plan is laid out for the tasks needed to be completed during the Design Project. The team had weekly stand-ups with the library staff.

8.1.1 Week 1

- Schedule weekly meetings with the library staff.
- Discussing potential solutions with the client.
- Doing research into smart technology use in identifying capacity.

8.1.2 Week 2

- Defining a list of requirements.
- Discussing with the client what priority certain requirements should have.
- Collected API data for smart plugs from the library database.
- Cleaned the data.
- Performed statistical analysis on smart plug data.
- Determined the threshold value for classifying smart plug to be in 'OFF' state.

8.1.3 Week 3

- Refined the user stories with the stakeholders.
- Organised Trello with user stories with MoSCoW prioritization.
- Accessed for desk motion sensor data from the API.
- Basic real-time data visualization of smart plug data.

8.1.4 Week 4

- Discussion with stakeholders on where to represent information such as occupancy levels and trend data.
- Allocation of user stories within teammates.
- Making functional and non-functional requirements.
- Cleaning for desk motion sensor data.
- Starting working on user stories.

8.1.5 Week 5

Work was done on the following requirements in week 5.

- The student should be able to check how long a monitor has been available for.
- The student should be able to check if a monitor seat is available or occupied.
- The librarian should receive a notification for inactivity.
- Formulating the business rule for determining what inactivity is.
- The alerts received by the librarian must contain useful information like the inactivity duration and the exact monitor reference number.
- The alerts must have a spam prevention system.
- The librarian should have the option of marking a monitor as defective.
- Determining how insights will be shown to the stakeholders.
- Making dummy data for simulating the behaviour of 'exam period' and 'Christmas breaks'.
- Integrating desk-monitor sensor with smart plug data to better under the occupancy.
- The visualizations for the trends should be interpretable for non-technical users.

8.1.6 Week 6

The following user stories were further divided into requirements.

- The data representation for seat availability must be accurate.
- The data on the online platform should not have a latency.
- The information for occupancy must be clear for the user to make a decision.
- The system shall implement robust access control mechanisms to restrict access to monitor area occupancy data to authorized personnel only, in accordance with privacy regulations.
- The system shall provide mechanisms for users to access, modify, or delete their personal data as per privacy regulations.
- The system shall ensure that all data collected, processed, and stored complies with relevant privacy regulations, including obtaining necessary consent from users where required.

8.1.7 Week 7

The following user stories were further divided into requirements.

- The monitor area occupancy detection system shall be seamlessly integrated with the library's existing management software to facilitate data sharing and interoperability.
- The integrated system shall automate the collection of monitor occupancy data from sensors and smart plugs, eliminating the need for manual data entry or manipulation.
- The integrated system shall provide real-time reporting capabilities within the library management software, allowing administrators to monitor the monitor area occupancy and usage patterns in real time.
- Library managers shall have the ability to customize occupancy thresholds based on factors such as user preferences, mobility challenges, and the need for longer breaks.
- Threshold adjustments shall be configurable through the system's administrative interface, ensuring flexibility and adaptability to changing requirements.
- The system shall incorporate user feedback mechanisms to gather input from individuals with diverse needs regarding the effectiveness of accommodations and services.
- Analysis of historical data shall inform proactive maintenance strategies and optimization of sensor deployment to maximize reliability and minimize downtime.

8.1.8 Week 8

In week 8, testing of all the user stories began. Usability testing and compatibility testing started.

- Library managers should have the ability to observe the energy consumption of each monitor during stand-by.
- Library managers should have the ability to observe the energy consumption of each monitor during live, normal usage.

8.1.9 Week 9

Same as week 8. Additional requirements were implemented here.

8.1.10 Week 10

The team worked on the presentation this week and final testing before deployment.

8.2 Responsibilities

The team was composed of people with various interests and skills. After the initial design meetings with the stakeholders and the division into requirements, each team member took the tasks that matched their interest and skill level in that segment. Roles sometimes interchanged and every member tried to help the rest of the team as much as possible to ensure that the final product is at the expected quality level. Overall, each member had their own input in both the design phase, as well as the front-end and back-end development of the project.

8.3 Team evaluation

Communication between team members went smoothly with the help of frequent meetings where any arising issues were discussed. Moreover, we used messaging platforms to further communicate about project work when we were not able to meet up. Everyone in the team was given a role they were enthusiastic about which kept the motivation levels high throughout the entire duration of the project. Overall, there were no arguments and disagreements as we tried to take every major decision as a team and consistently helped each other when needed.

8.4 Final result

The final result was a complete system for both librarians and students to use. In addition, the monitor occupancy display was also integrated into Mazemap, a feat no other student team had done so far. The library staff were satisfied with the results and are looking forward to using the system not only for managing the monitor area but also for achieving sustainability goals regarding replacing any monitors drawing excessive amounts of power. The final product aligns with the client's preferences and contains additional functionalities they requested throughout the process.

8.5 Conclusion

Through this project, we aimed to offer a solution to the overcrowding problem at the library by implementing a system that helps display and manage the available seats such that no student travels to the library only to find that the monitor area is full. By using real-time sensors and smart plug data the system offers an accurate representation of the monitor occupancy. An overview of the available seats can be found on a page dedicated to the students. A more advanced dashboard was implemented for library staff such that they can observe trends regarding the monitor occupancy, power consumption and overall business in the library monitor area. Moreover, they can alert students and mark monitors as defective when needed.

Customer satisfaction was high as all of the features they requested were implemented. The feedback received from conducting weekly meetings with the library staff helped us tailor the design to their preferences and discuss changes when needed. By testing the Mazemap interface with fellow students we were able to verify its usability and make sure it is intuitive enough to navigate through.

Handling sensor data posed challenges such as inaccurate values and wrong sensor IDs. There were also unpredictable factors such as monitor brightness that were discovered after thorough examinations of the monitor areas. Nevertheless, the final solution includes a fix for all of the problems mentioned and was extensively tested.

Possible enhancements to the system include broadening the scope of the solution such that it is utilized in other university areas and used to lower power consumption. Improving the user interface by adding a more colourblind-friendly palette was not a priority for this project but could be beneficial in the future and improve accessibility.

The delivered solution has the potential to set the groundwork for future innovations at the University of Twente. The library staff are looking forward to utilizing the implementation to enhance the student academic experience and possibly expand it in the future with the features we recommended. The knowledge and experience gained through this process are invaluable assets for the future careers of all team members.

Bibliography

[1] Ashley Aitken and Vishnu Ilango. A comparative analysis of traditional software engineering and agile software development. In *2013 46th Hawaii International Conference on System Sciences*, pages 4751–4760, 2013. doi:10.1109/HICSS.2013.31.

[2] Eduardo Almeida, Jorge Mascena, Ana Paula Furtado, Alexandre Alvaro, Vinicius Garcia, Silvio Meira, and Daniel Lucrédio. The domain analysis concept revisited: A practical approach. pages 43–57, 06 2006. doi:10.1007/11763864_4.

[3] Barry W. Boehm. Seven basic principles of software engineering. *Journal of Systems and Software*, 3(1):3–24, 1983. URL: https://www.sciencedirect.com/science/article/pii/0164121283900031, doi:10.1016/0164-1212(83)90003-1.

[4] Antonius Gradus Johannes Jansen. *Architectural design decisions*. PhD thesis, 2008. Relation: http://www.rug.nl/ date$_{s}ubmitted : 2008Rights : UniversityofGroningen$.

[5] Tatiana Kravchenko, Tatiana Bogdanova, and Timofey Shevgunov. Ranking requirements using moscow methodology in practice. In Radek Silhavy, editor, *Cybernetics Perspectives in Systems*, pages 188–199, Cham, 2022. Springer International Publishing.